**Debugging in Turbo C++ IDE**

Turbo C++ provides many useful debugging features. The debugging features of turbo C++ are available in Debug menu. Some important debugging features of turbo C++ are as follows:

**Single stepping**

A program may not execute even if it has been compiled successfully. The debugger provides the facility to find errors by executing one line of the program at one point. It enables the programmer to detect the exact place of error. The following procedure is used to execute one step at a time with single stepping:

Select Run > Trace Into OR press F7 key.

**Watches**

The watch or watch expression is used to check the value of variable as the program executes. It indicates how the values of variables change during program execution. It is normally used in combination with single stepping.

The following procedure is used to use watch expression:

* Place the cursor on the variable whose value is to be checked.
* Select Debug>Watches. A submenu will appear.
* Select ADD Watch from the submenu. Or press CTRL + F7. A dialog box will appear. The selected variable will appear in Watch expression field.
* Click OK or press enter. A new window will appear indicating that the selected symbol is undefined.
* Select Run > Trace Into Or press F7 to execute single stepping. The values of the selected variable will appear in watch window.

**Breakpoints**

A breakpoint is used to mark a part of program where program execution will stop. The program executes all statements up to the break point and then stops. The user can check the values of a variable at this point by using Watch window by single stepping the remaining part of the program.

The following procedure is used to insert a breakpoint in the program.

* Place the cursor on the line on which the breakpoint is to be inserted.
* Select Debug > Toggle breakpoint Or press CTRL + F8. The breakpoint will be inserted and the line will be highlighted.

**Evaluate/Modify Window**

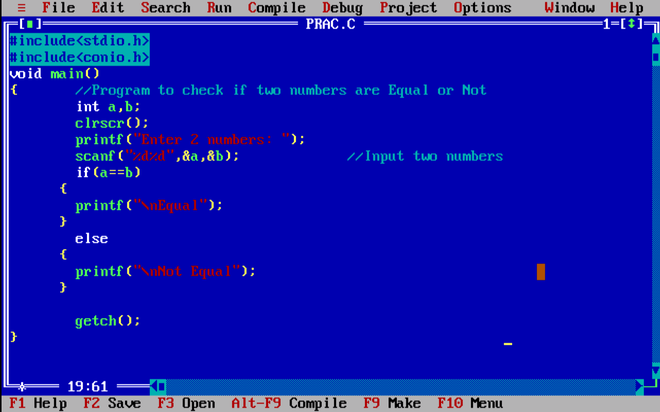
 The evaluate/Modify window is used to change the value of variable during program execution. It can be useful if the user is single stepping the program and wants to change the value of a certain variable. The following procedure is used to evaluate/modify window:

* Select Debug > Evaluate/modify. A new window will appear with three fields.
* Enter the name of the variable whose value is to be modified in Expression field.
* Enter the new value for the variable in New Value field. The value of the third field Result will also change automatically.

**Debugging in Turbo C++**

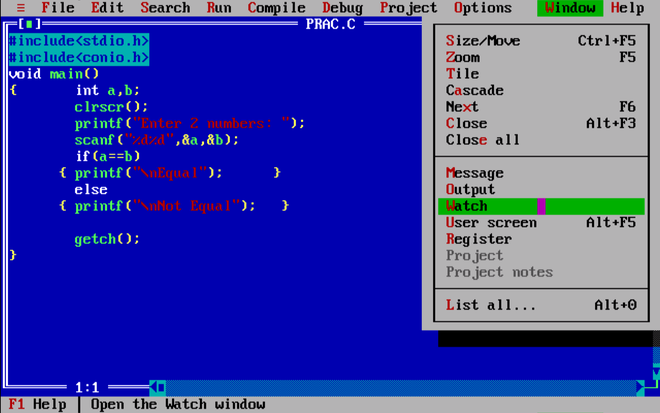
Debugging is a process in which the programmers detect and remove the existing or potential errors/bugs in their programs. It is an excellent practice for writing an effective code. Now, Turbo C++ provides a variety of useful features for the sake of its community. Some options that play an important role in our debugging are Watch and Trace. Let’s understand these concepts by taking the following program:

**Example:**Write a program to check if two numbers are Equal or Not

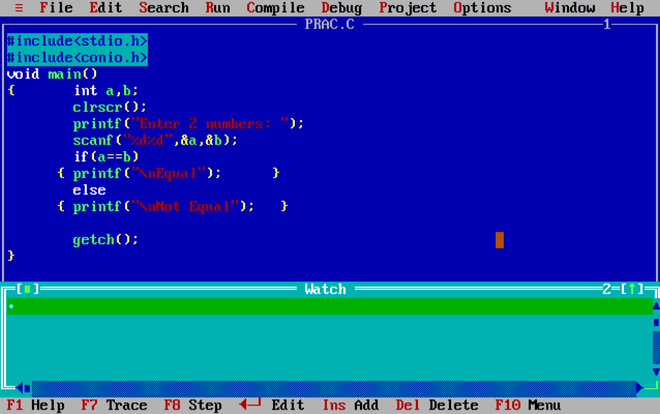


Now go through the undermentioned steps to get a clear idea of how to debug:

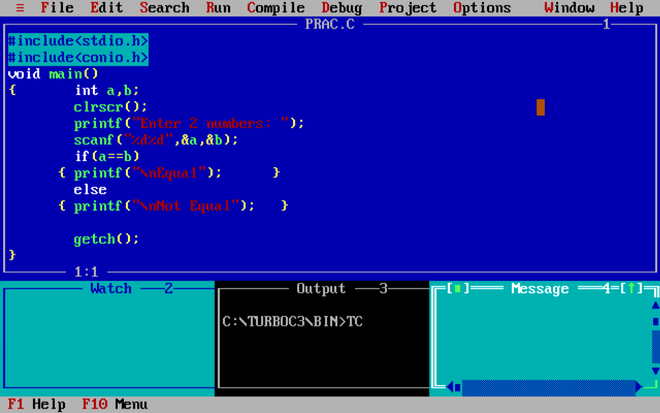
**Step 1:**Go to Window Menu and click *Watch* under it.



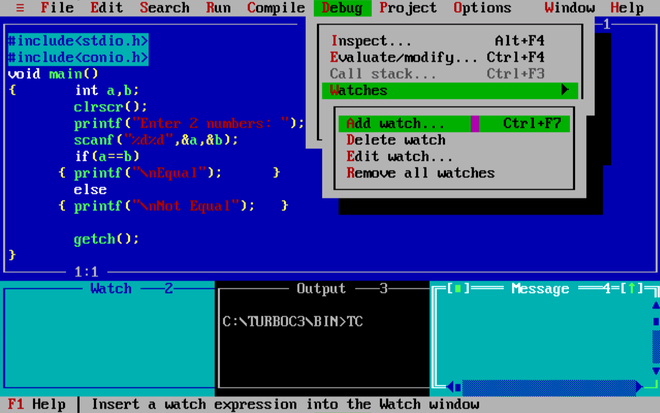
**Step 2:**You’ll be able to see a panel named Watch at the bottom of your IDE.



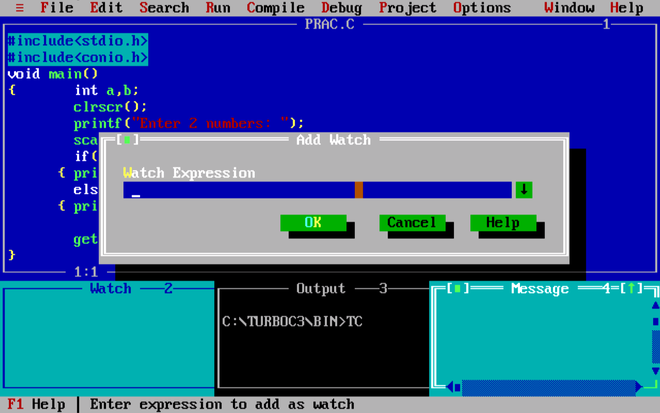
**Step 3:**Similarly, from the Window menu we’ll select *Output* and *Message* and then click on Tile for a better view.



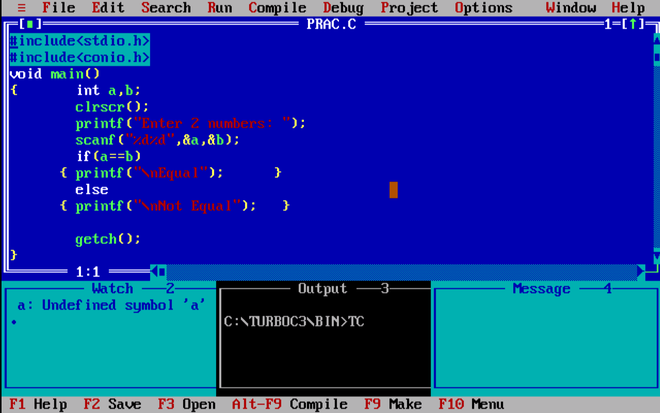
**Step 4:** Go to *Debug*, then under Watches choose *Add Watch (or Ctrl+F7).*



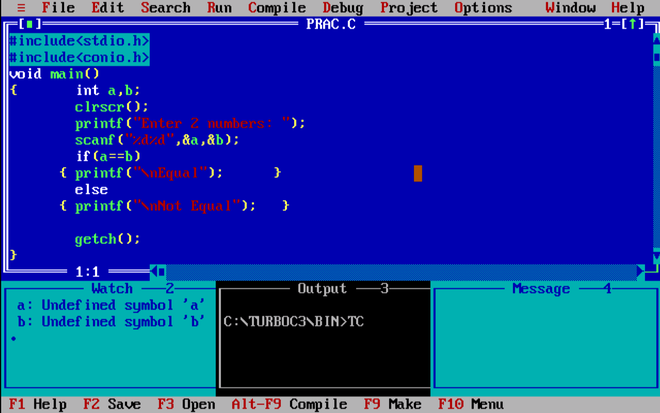
**Step 5:**A dialog box will appear on your screen saying “*Enter Expression to add as Watch*”. Enter ‘a’.



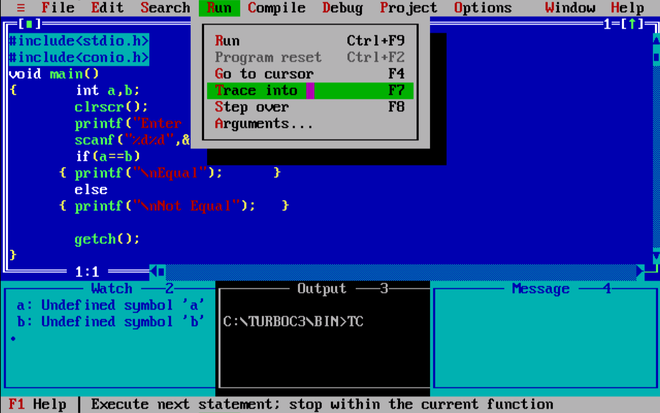
**Step 6:**Watch panel will show the result of adding ‘a’ as a watch expression.



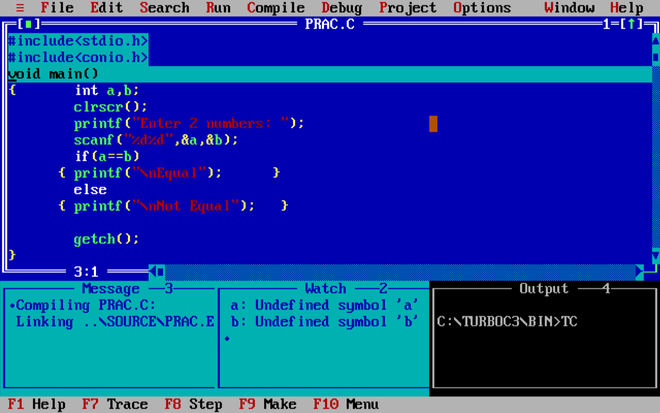
**Step 7:**Follow Step 5 again and Enter ‘b’.



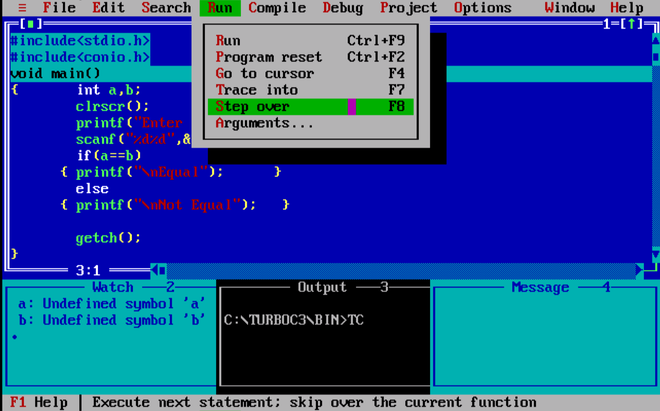
**Step 8:**Go to Run Option and Click *Trace Into (or F7).*



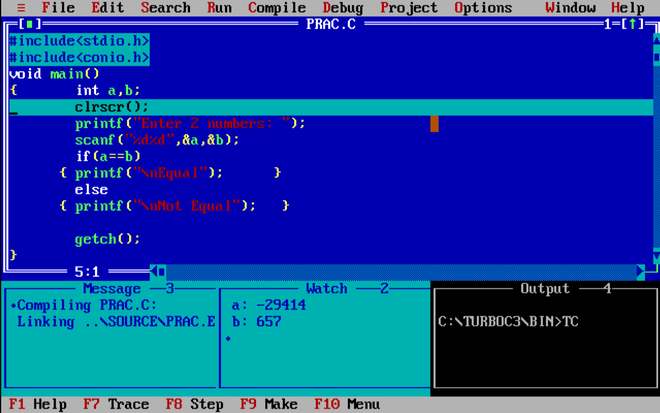
**Step 9:**You will notice void main() being highlighted. This means our program has started running.



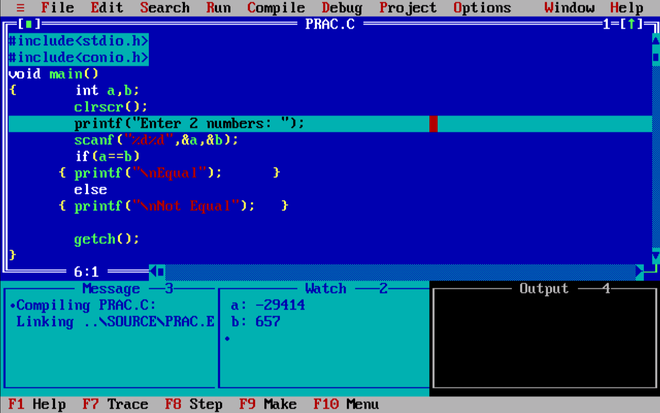
**Step 10:**Go to the Run menu and Click *Step Over (or F8)* to proceed to the next step.



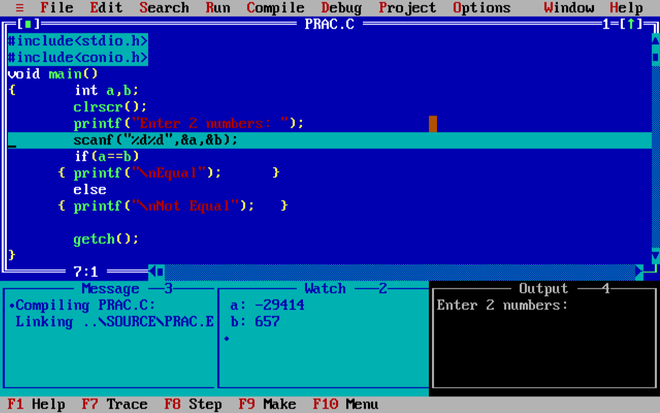
**Step 11:** Step over proceeds to the next statement.



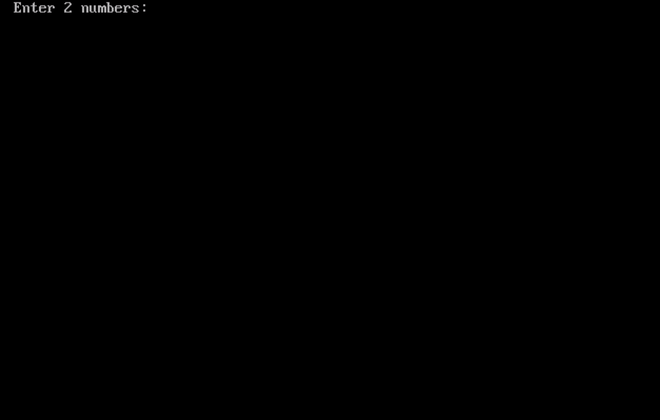
**Step 12:** Press F8 to execute the next statement.



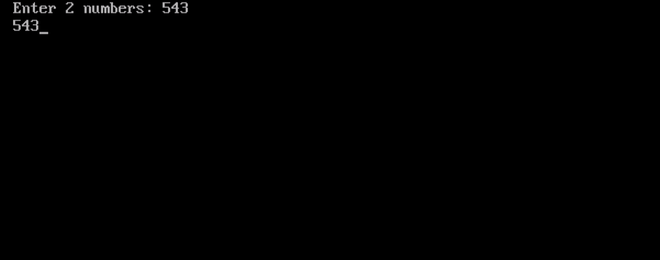
**Step 13:**Step Over (F8).



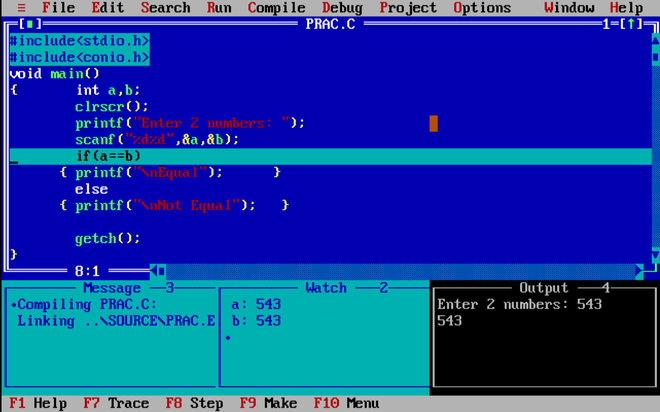
**Step 14:**Proceed to Next Step (F8).



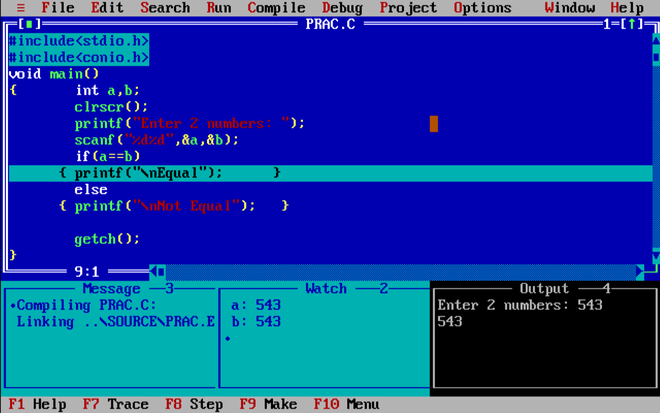
**Step 15:**Enter the 2 numbers in the console.



**Step 16:**Take a look at the watch and output panels. After getting the input from the user, the compiler overwrites the values in the variables.

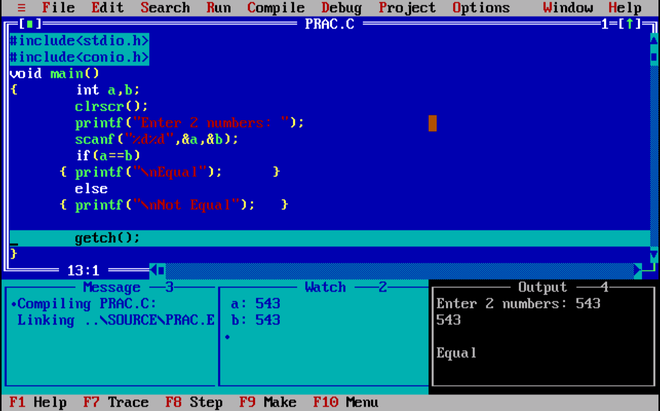


**Step 17:**Press F8.

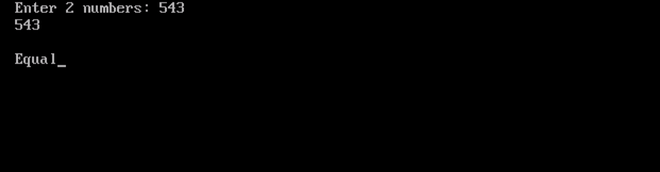


Remember since our *if the condition* is True the compiler shouldn’t check the *else statement*.

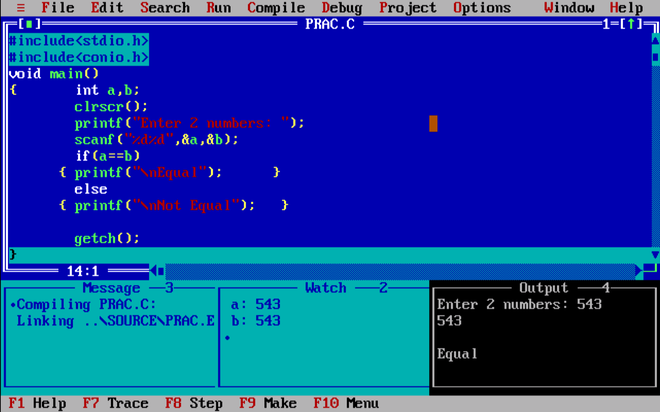
**Step 18:**Press F8 to proceed to the next step.



**Step 19:**Press F8 to check what is the next step. You’ll be able to see the Output screen now.



**Step 20:** Press F8 to check what will happen next.



Notice the closing bracket is highlighted now, which means the program has ended. If F8 is pressed again the program will run over again from the first step.

Online Notes

[https://www.geeksforgeeks.org/debugging-in-turbo-cpp/#:~:text=Step%201%3A%20Go%20to%20Window,(or%20Ctrl%2BF7).](https://www.geeksforgeeks.org/debugging-in-turbo-cpp/%23:~:text=Step%201%3A%20Go%20to%20Window,(or%20Ctrl%2BF7).)